

**Schulinterner Lehrplan
zum Kernlehrplan für die gymnasiale Oberstufe
(Sekundarstufe II)**

**gültig ab Schuljahr 2014/15
(letzte Überarbeitung: 08/2017)**

Informatik

Inhalt

1 Rahmenbedingungen der fachlichen Arbeit.....	3
2 Entscheidungen zum Unterricht	5
2.1 Unterrichtsvorhaben.....	5
2.1.1 Übersichtsraster Unterrichtsvorhaben Einführungsphase	6
2.1.2 Konkretisierte Unterrichtsvorhaben Einführungsphase	9
2.1.3 Übersichtsraster Unterrichtsvorhaben Qualifikationsphase.....	22
2.1.4 Konkretisierte Unterrichtsvorhaben Qualifikationsphase.....	25
2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit.....	42
2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung.....	43
2.3.1 Beurteilungsbereich Klausuren	43
2.3.2 Beurteilungsbereich Sonstige Mitarbeit.....	44
2.4 Lehr- und Lernmittel	46
3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen	47
4 Qualitätssicherung und Evaluation	48

1 Die Rahmenbedingungen der fachlichen Arbeit

Bei der FHS handelt es sich um eine allgemein vierzügige Schule in der Stadt Herdecke mit zurzeit ca. 900 Schülerinnen und Schülern, ca. 60 Lehrerinnen und Lehrern. Das Einzugsgebiet der Schule umfasst im Wesentlichen die Stadt Herdecke. Im Bereich der Sekundarstufe II kooperiert die FHS in einigen Fächern mit dem Gymnasium in Wetter.

Das Fach Informatik wird an der FHS ab der Jahrgangsstufe 8 im Differenzierungsbereich zweistündig in ein oder zwei Kursen, je nach Anwahlverhalten durch die Schülerinnen und Schüler, unterrichtet. In der zweijährigen Laufzeit dieser Kurse baut der Kurs in altersstufengerechter Weise unter anderem die Grundlagen der Algorithmik am Beispiel von Scratch auf. Ferner recherchieren Die Schülerinnen und Schüler der Klasse 8 grundlegende Begriffe der Informatik präsentieren diese mithilfe von Standardsoftware. In der Klasse 9 wird der Umgang mit Scratch wiederholt und vertieft und außerdem die Textverarbeitung in Richtung großer Dokumente im Rahmen von Praktikums- und Facharbeiten behandelt. Weitere Themengebiete des Differenzierungskurses sind das Bauen und Programmieren von LEGO-Mindstorms Robotern, die Einführung in die Kryptologie, das Programmieren von Android Apps oder das Erstellen Html-basierter Webseiten.

Das Ziel der Fachgruppe Informatik ist es, in den Jahrgangsstufen der Mittelstufe einen für alle verpflichtenden Kurs zum Umgang mit informatischen Systemen durchzuführen, eine sogenannte *Informationstechnische Grundlagenbildung*, die jedoch nicht unmittelbar dem Fach Informatik zuzuordnen ist. Im Schuljahr 16/17 konnte dieses Ziel erstmals für die Jahrgangsstufen 5 und 7 im Umfang eines Quartals realisiert werden.

In der Sekundarstufe II bietet die FHS für die Schülerinnen und Schüler in allen Jahrgangsstufen Informatik an. Um insbesondere Schülerinnen und Schülern gerecht zu werden, die in der Sekundarstufe I keinen Informatikunterricht besucht haben, wird in Kursen der Einführungsphase besonderer Wert darauf gelegt, dass keine Vorkenntnisse aus der Sekundarstufe I zum erfolgreichen Durchlaufen des Kurses erforderlich sind.

Der Unterricht der Sekundarstufe II wird mit Hilfe der Programmiersprache Java durchgeführt. In der Einführungsphase kommt dabei die didaktische Lernumgebung Greenfoot zum Einsatz, welche das Erstellen von anschaulichen Programmen erleichtert. Die Umgebung ist Freeware und kann für alle gängigen Betriebssysteme unter www.greenfoot.org heruntergeladen werden. Nach der Einarbeitungsphase wird auf die komplexere Lernumgebung Java Editor umgestellt, welche ebenfalls kostenfrei erhältlich ist.

Es werden die aktuellen Schulbücher „Informatik 1“ und „Informatik 2“ vom Schöningh Verlag eingesetzt.

Durch projektartiges Vorgehen, offene Aufgaben und Möglichkeiten, Problemlösungen zu verfeinern oder zu optimieren, entspricht der Informatikunterricht der Oberstufe in besonderem Maße den Erziehungszielen, Leistungsbereitschaft zu fördern, ohne zu überfordern.

Die gemeinsame Entwicklung von Materialien und Unterrichtsvorhaben, die Evaluation von Lehr- und Lernprozessen sowie die stetige Überprüfung und eventuelle Modifikation des schulinternen Curriculums durch die Fachkonferenz

Informatik stellen einen wichtigen Beitrag zur Qualitätssicherung und -entwicklung des Unterrichts dar.

Zurzeit besteht die Fachschaft Informatik der FHS aus zwei Lehrkräften. Frau Hesse hat den Vorsitz inne und Herr Herberg fungiert als Stellvertreter. Im Schulgebäude stehen zwei Computerräume mit 15 bzw. 28 Computerarbeitsplätzen und ein Selbstlernzentrum (m@z) mit 6 Plätzen zur Verfügung. Die Schülerinnen und Schüler können die Computer im m@z zur Recherche im Internet oder zur Bearbeitung schulischer Aufgaben verwenden. Der Unterricht erfolgt im 90-Minuten-Takt.

2 Entscheidungen zum Unterricht

2.1 Unterrichtsvorhaben

Die Darstellung der Unterrichtsvorhaben im schulinternen Lehrplan besitzt den Anspruch, sämtliche im Kernlehrplan angeführten Kompetenzen abzudecken. Dies entspricht der Verpflichtung jeder Lehrkraft, Schülerinnen und Schülern Lerngelegenheiten zu ermöglichen, so dass alle Kompetenzerwartungen des Kernlehrplans von ihnen erfüllt werden können.

Die entsprechende Umsetzung erfolgt auf zwei Ebenen: der Übersichts- und der Konkretisierungsebene.

Im „Übersichtsraster Unterrichtsvorhaben“ (Kapitel 2.1.1) wird die für alle Lehrerinnen und Lehrer gemäß Fachkonferenzbeschluss verbindliche Verteilung der Unterrichtsvorhaben dargestellt. Das Übersichtsraster dient dazu, den Kolleginnen und Kollegen einen schnellen Überblick über die Zuordnung der Unterrichtsvorhaben zu den einzelnen Jahrgangsstufen sowie den im Kernlehrplan genannten Kompetenzen, Inhaltsfeldern und inhaltlichen Schwerpunkten zu verschaffen. Der ausgewiesene Zeitbedarf versteht sich als grobe Orientierungsgröße, die nach Bedarf über- oder unterschritten werden kann. Um Freiraum für Vertiefungen, besondere Schülerinteressen, aktuelle Themen bzw. die Erfordernisse anderer besonderer Ereignisse (z.B. Praktika, Kursfahrten o.ä.) zu erhalten, wurden im Rahmen dieses schulinternen Lehrplans ca. 75 Prozent der Bruttounterrichtszeit verplant.

Während der Fachkonferenzbeschluss zum „Übersichtsraster Unterrichtsvorhaben“ zur Gewährleistung vergleichbarer Standards sowie zur Absicherung von Lerngruppenübertritten und Lehrkraftwechseln für alle Mitglieder der Fachkonferenz Bindekraft entfalten soll, beinhaltet die Ausweisung „konkretisierter Unterrichtsvorhaben“ (Kapitel 2.1.2) Beispiele und Materialien, die empfehlenden Charakter haben. Referendarinnen und Referendaren sowie neuen Kolleginnen und Kollegen dienen diese vor allem zur standardbezogenen Orientierung in der neuen Schule, aber auch zur Verdeutlichung von unterrichtsbezogenen fachgruppeninternen Absprachen zu didaktisch-methodischen Zugängen, fächerübergreifenden Kooperationen, Lernmitteln und -orten sowie vorgesehenen Leistungsüberprüfungen, die im Einzelnen auch den Kapiteln 2.2 bis 2.3 zu entnehmen sind.

Da in den folgenden Unterrichtsvorhaben Inhalte in der Regel anhand von Problemstellungen in Anwendungskontexten bearbeitet werden, werden in einigen Unterrichtsvorhaben jeweils mehrere Inhaltsfelder angesprochen.

2.1.1 Übersichtsraster Unterrichtsvorhaben Einführungsphase

Einführungsphase	
<u>Unterrichtsvorhaben E-I</u> Thema: <i>Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten</i> Zentrale Kompetenzen: •Argumentieren •Darstellen und Interpretieren •Kommunizieren und Kooperieren Inhaltsfelder: •Informatiksysteme •Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: •Einzelrechner •Dateisystem •Internet •Einsatz von Informatiksystemen Zeitbedarf: 6 Stunden	<u>Unterrichtsvorhaben E-II</u> Thema: <i>Grundlagen der objektorientierten Analyse, Modellierung und Implementierung</i> Zentrale Kompetenzen: •Modellieren •Implementieren •Darstellen und Interpretieren •Kommunizieren und Kooperieren Inhaltsfelder: •Daten und ihre Strukturierung •Formale Sprachen und Automaten Inhaltliche Schwerpunkte: •Objekte und Klassen •Syntax und Semantik einer Programmiersprache Zeitbedarf: 8 Stunden

Einführungsphase	
<u>Unterrichtsvorhaben E-III</u> Thema: <i>Grundlagen der objektorientierten Programmierung und algorithmischer Grundstrukturen in Java - Greenfoot</i> Zentrale Kompetenzen: •Argumentieren •Modellieren •Implementieren •Kommunizieren und Kooperieren Inhaltsfelder: •Daten und ihre Strukturierung •Algorithmen •Formale Sprachen und Automaten Inhaltliche Schwerpunkte: •Objekte und Klassen •Syntax und Semantik einer Programmiersprache •Analyse, Entwurf und Implementierung einfacher Algorithmen Zeitbedarf: 20 Stunden	<u>Unterrichtsvorhaben E-IV</u> Thema: <i>Modellierung und Implementierung von Klassen- und Objektbeziehungen im Java Editor und Entwicklung einer GUI</i> Zentrale Kompetenzen: •Argumentieren •Modellieren •Implementieren •Darstellen und Interpretieren •Kommunizieren und Kooperieren Inhaltsfelder: •Daten und ihre Strukturierung •Algorithmen •Formale Sprachen und Automaten Inhaltliche Schwerpunkte: •Objekte und Klassen •Syntax und Semantik einer Programmiersprache •Analyse, Entwurf und Implementierung einfacher Algorithmen Zeitbedarf: 18 Stunden

Einführungsphase	
<u>Unterrichtsvorhaben E-V</u> Thema: <i>Such- und Sortialgorithmen anhand kontextbezogener Beispiele</i> Zentrale Kompetenzen: •Argumentieren •Modellieren •Darstellen und Interpretieren •Kommunizieren und Kooperieren Inhaltsfelder: •Algorithmen Inhaltliche Schwerpunkte: •Algorithmen zum Suchen und Sortieren •Analyse, Entwurf und Implementierung einfacher Algorithmen Zeitbedarf: 10 Stunden	<u>Unterrichtsvorhaben E-VI</u> Thema: <i>Leben in einer digitalen Welt- Immer mehr Möglichkeiten, immer mehr Gefahren?</i> Zentrale Kompetenzen: •Argumentieren •Darstellen und Interpretieren •Kommunizieren und Kooperieren Inhaltsfelder: •Informatik, Mensch und Gesellschaft •Informatiksysteme Inhaltliche Schwerpunkte: •Wirkungen der Automatisierung •Geschichte der automatischen Datenverarbeitung •Binäre Codierung und Verarbeitung •Digitalisierung Zeitbedarf: 15 Stunden
Summe Einführungsphase: 77	

2.1.2 Konkretisierte Unterrichtsvorhaben Einführungsphase

Die folgenden Kompetenzen aus dem Bereich *Kommunizieren und Kooperieren* werden in allen Unterrichtsvorhaben der Einführungsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden Fachausdrücke bei der Kommunikation über informatische Sachverhalte (K),
- präsentieren Arbeitsabläufe und -ergebnisse (K),
- kommunizieren und kooperieren in Gruppen und in Partnerarbeit (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung und gemeinsamen Verwendung von Daten unter Berücksichtigung der Rechteverwaltung (K)

Unterrichtsvorhaben EF-I

Thema: Einführung in die Nutzung von Informatiksystemen und in grundlegende Begrifflichkeiten

Leitfragen: Womit beschäftigt sich die Wissenschaft der Informatik? Wie kann die in der Schule vorhandene informatische Ausstattung genutzt werden?

Vorhabenbezogene Konkretisierung:

Das erste Unterrichtsvorhaben stellt eine allgemeine Einführung in das Fach Informatik dar. Dabei ist zu berücksichtigen, dass für manche Schülerinnen und Schüler in der Einführungsphase der erste Kontakt mit dem Unterrichtsfach Informatik stattfindet, so dass zu Beginn Grundlagen des Fachs behandelt werden müssen. Schülerinnen und Schüler, die schon im Differenzierungsbereich Informatik hatten, werden in dieser Sequenz unterstützend in die Unterrichtsdurchführung einbezogen.

Zunächst wird auf den Begriff der Information eingegangen. Des Weiteren soll der grundlegende Aufbau eines Rechnersystems im Sinne der Von-Neumann-Architektur erarbeitet werden und mit dem grundlegenden Prinzip der Datenverarbeitung (Eingabe-Verarbeitung-Ausgabe) in Beziehung gesetzt werden.

Anschließend wird auf die Übertragung von Daten im Sinne des Sender-Empfänger-Modells eingegangen. Dabei wird eine überblicksartige Vorstellung der Kommunikation von Rechnern in Netzwerken erarbeitet.

Bei der Beschäftigung mit Datenübermittlung und Datenverarbeitung ist ein Bezug zur konkreten Nutzung der informatischen Ausstattung der Schule herzustellen. So wird in die verantwortungsvolle Nutzung dieser Systeme eingeführt.

Zeitbedarf: 6 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Informatik, Information und Speicherung (a) Informatik als Wissenschaft der Verarbeitung von Informationen (b) Speichern von Daten mit informatischen Systemen (Ordnerstruktur, Dateibezeichnung usw.) und Vereinbarung von Richtlinien zur Datenspeicherung	Die Schülerinnen und Schüler • beschreiben und erläutern den Aufbau und die Arbeitsweise singulärer Rechner am Beispiel der „Von-Neumann-Architektur“ (A) • bewerten anhand von Beispielen die Auswirkungen des Einsatzes von Informatiksystemen (A) • nutzen die im Unterricht eingesetzten Informatiksysteme selbstständig, sicher, zielführend und verantwortungsbewusst (D) • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation (K).	<i>Materialien:</i> USB-Stick, Lernplattform itslearning, Smartphone, Navigationsgerät
2. Aufbau informatischer Systeme (a) Identifikation typischer Komponenten informatischer Systeme und anschließende Beschränkung auf das Wesentliche, Herleitung der „Von-Neumann-Architektur“ (b) Identifikation des EVA-Prinzips (Eingabe-Verarbeitung-Ausgabe) als Prinzip der Verarbeitung von Daten und Grundlage der „Von-Neumann-Architektur“		<i>Material:</i> Demonstrationshardware Anhand eines Demonstrationsrechners entdecken Schülerinnen und Schüler die verschiedenen Hardwarekomponenten eines Informatiksystems. <i>Medien:</i> Freemind
3. Informations- und Datenübermittlung in Netzen (a) „Sender-Empfänger-Modell“ und seine Bedeutung für die Eindeutigkeit von Kommunikation (b) Informatische Kommunikation in Rechnernetzen (c) Grundlagen der technischen Umsetzung von Rechnerkommunikation am Beispiel des Internets (d) Richtlinien zum verantwortungsvollen Umgang mit dem Internet		<i>Beispiel:</i> Netzwerkplan der Schule <i>Medien:</i> Freemind

Unterrichtsvorhaben EF-II (8h)

Thema: Grundlagen der objektorientierten Analyse, Modellierung und Implementierung anhand eines Roboter-Szenarios

Leitfrage: *Wie lassen sich Gegenstandsbereiche informatisch modellieren und im Sinne einer Simulation informatisch realisieren?*

Vorhabenbezogene Konkretisierung:

Ein zentraler Bestandteil des Informatikunterrichts der Einführungsphase ist die Objektorientierte Programmierung. Dieses Unterrichtsvorhaben führt in die Grundlagen der Analyse, Modellierung und Implementierung in diesem Kontext ein.

Dazu werden zunächst konkrete Gegenstandsbereiche aus der Lebenswelt der Schülerinnen und Schüler analysiert und im Sinne des Objektorientierten Paradigmas strukturiert. Dabei werden die grundlegenden Begriffe der Objektorientierung (Objekt, Klasse, Vererbung) und Modellierungswerkzeuge wie Objektkarten, Klassenkarten oder Beziehungsdiagramme eingeführt.

Im Anschluss wird mit der Realisierung erster Projekte mit Hilfe der didaktischen Programmierumgebung GREENFOOT begonnen. Die von der Bibliothek vorgegebenen Klassen werden von Schülerinnen und Schülern in Teilen analysiert und entsprechende Objekte anhand einfacher Problemstellungen erprobt. Die Lernenden implementieren und testen einfache Programme. Die Greenfoot-Umgebung ermöglicht es, Beziehungen zwischen Klassen zu einem späteren Zeitpunkt zu thematisieren. So kann der Fokus hier auf Grundlagen wie der Unterscheidung zwischen Klasse und Objekt, Attribute, Methoden, Objektidentität und Objektzustand gelegt werden. Dazu muss der grundlegende Aufbau einer Java-Klasse thematisiert werden.

Da bei der Umsetzung dieser ersten Projekte konsequent auf die Verwendung von Kontrollstrukturen verzichtet wird und der Quellcode aus einer rein linearen Sequenz besteht, ist auf diese Weise eine Fokussierung auf die Grundlagen der Objektorientierung möglich, ohne dass algorithmische Probleme ablenken. Natürlich kann die Arbeit an diesen Projekten unmittelbar zum nächsten Unterrichtsvorhaben führen. Dort stehen unter anderem Kontrollstrukturen im Mittelpunkt.

Zeitbedarf: 8 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Identifikation von Objekten (a) Am Beispiel eines lebensweltnahen Beispiels werden Objekte im Sinne der Objektorientierten Modellierung eingeführt. (b) Objekte werden mit Objektkarten visualisiert und mit sinnvollen Attributen und „Fähigkeiten“, d.h. Methoden versehen. (c) Manche Objekte sind prinzipiell typgleich und werden so zu einer Klasse zusammengefasst. (d) Vertiefung: Modellierung weiterer Beispiele ähnlichen Musters	Die Schülerinnen und Schüler • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), • stellen den Zustand eines Objekts dar (D).	<i>Beispiel:</i> Autos, Pferde <i>Materialien:</i> Lehrbuch S.20
2. Analyse von Objekten und Klassen in Greenfoot (a) Objektorientierte Programmierung als modularisiertes Vorgehen (Entwicklung von Problemlösungen auf Grundlage vorhandener Klassen) (b) Teilanalyse und Erprobung der Klassen eines Greenfoot-Szenarios		<i>Materialien:</i> Rover-Szenario für Greenfoot oder Wombat, little Crab
3. Implementierung linearer Handlungsabläufe (a) Grundaufbau einer Java-Klasse (b) Deklaration, Initialisierung von Objekten (c) Implementation eigener Methoden (d) Speicherung von Welten		<i>Beispiele:</i> Aufgaben Buch Seite 27

Unterrichtsvorhaben EF-III (20h)

Thema: Algorithmische Grundstrukturen in Java

Leitfragen: *Wie lassen sich Aktionen von Objekten flexibel realisieren?*

Vorhabenbezogene Konkretisierung:

Das Ziel dieses Unterrichtsvorhabens besteht darin, das Verhalten von Objekten flexibel zu programmieren. Ein erster Schwerpunkt liegt auf der Entwicklung mehrerer nicht linearer Projekte. Zunächst wird ein Projekt bearbeitet, das in Anlehnung an das vorangegangene Unterrichtsvorhaben eine Szene darstellt, die lediglich aus Objekten besteht, zu denen Greenfoot Klassen vorgibt. Für die Umsetzung dieses Projekts werden Kontrollstrukturen in Form von Schleifen und Verzweigungen benötigt und eingeführt.

Sind an einem solchen Beispiel im Schwerpunkt Schleifen und Verzweigungen eingeführt worden, sollen diese Konzepte an weiteren Beispielprojekten eingeübt werden. Bei der Lösung komplexerer Probleme gilt es dann, die bisher erlernten Sprachkonzepte und -methoden miteinander zu kombinieren.

Ein zweiter Schwerpunkt des Unterrichtsvorhabens liegt auf dem Einsatz von Variablen. Beginnend mit lokalen Variablen, die in Methoden und Zählschleifen zum Einsatz kommen, über Variablen in Form von Parametern und Rückgabewerten von Methoden, bis hin zu Variablen, die die Attribute einer Klasse realisieren, lernen die Schülerinnen und Schüler die unterschiedlichen Einsatzmöglichkeiten des Variablenkonzepts anzuwenden. Die Arbeit mit Methoden, Variablen und Attributen dient der sinnvollen Strukturierung und Erhöhung der Wiederverwendbarkeit größerer Softwareprojekte.

Komplexere Beziehungen zwischen Klassen werden in diesem Unterrichtsvorhaben zunächst nicht behandelt. Sie stellen den Schwerpunkt des folgenden Vorhabens dar.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1.Problemlösen am mit Hilfe einfacher Algorithmen (a) Bedingte Anweisung – Verzweigung (b) Verschachtelungen (c) Logische Verknüpfungen (d) Zählschleife (e) Bedingte Schleifen (vor- und nachprüfend)	Die Schülerinnen und Schüler - analysieren und erläutern einfache Algorithmen und Programme (A), - entwerfen einfache Algorithmen und stellen sie umgangssprachlich und grafisch dar (M), - ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), - modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen und Objekttypen zu (M), - ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), - modifizieren einfache Algorithmen und Programme (I), - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), - implementieren Algorithmen unter Verwendung von Variablen und Wertzuweisungen, Kontrollstrukturen sowie Methodenaufrufen (I), - implementieren einfache Algorithmen unter Beachtung der Syntax und Semantik einer Programmiersprache (I), - testen Programme schrittweise anhand von Beispielen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I).	<i>Beispiele:</i> Buch Seite 39 ff., 45 ff., 50 ff., 55ff.
2.Entwurf und Implementation komplexerer Algorithmen - Methoden (a) Entwicklung (b) Visualisierung (c) Implementation - Testen - Korrektur, Optimierung		<i>Beispiel:</i> Buch Seite 62 ff.
3.Strukturierung von Programmen (a) Modellierung und Implementierung eigener Methoden mit und ohne Parameterübergabe und/oder Rückgabewert (b) Attribute und Variablen (einfache Datentypen) (c) Konstruktoren (d) Geheimnisprinzip (e) Vertiefung: Weitere Projekte		<i>Beispiele:</i> Buch Seite 76, 84f., 89ff., 92ff.

Unterrichtsvorhaben EF-IV (18h)

Thema: Modellierung und Implementierung von Klassen- und Objektbeziehungen anhand von grafischen Spielen und Simulationen

Leitfrage: *Wie werden realistische Systeme anforderungsspezifisch reduziert, als Entwurf modelliert und implementiert? Wie kommunizieren Objekte und wie wird dieses dargestellt und realisiert?*

Vorhabenbezogene Konkretisierung:

Das Unterrichtsvorhaben hat die Entwicklung von Objekt -und Klassenbeziehungen zum Schwerpunkt. Dazu werden, ausgehend von der Realität, über Objektidentifizierung und Entwurf bis hin zur Implementation kleine Softwareprodukte in Teilen oder ganzheitlich erstellt.

Zuerst identifizieren die Schülerinnen und Schüler Objekte und stellen diese dar. Aus diesen Objekten werden Klassen und ihre Beziehungen in Entwurfsdiagrammen erstellt.

Nach diesem ersten Modellierungsschritt werden über Klassendokumentationen und der Darstellung von Objektkommunikationen anhand von Sequenzdiagrammen Implementationsdiagramme entwickelt. Danach werden die Implementationsdiagramme unter Berücksichtigung der Klassendokumentationen in Javaklassen programmiert. In einem letzten Schritt wird das Konzept der Vererbung sowie seiner Vorteile erarbeitet. Dazu werden die wichtigsten Varianten der Vererbung anhand von verschiedenen Projekten vorgestellt. Zunächst wird die Vererbung als Spezialisierung im Sinne einer einfachen Erweiterung einer Oberklasse vorgestellt. Darauf folgt ein Projekt, welches das Verständnis von Vererbung um den Aspekt der späten Bindung erweitert, indem Dienste einer Oberklasse überschrieben werden.

Schließlich sind die Schülerinnen und Schüler in der Lage, eigene kleine Softwareprojekte zu entwickeln. Dabei können arbeitsteilige Vorgehensweisen zum Einsatz kommen. In diesem Zusammenhang wird auch das Erstellen von graphischen Benutzeroberflächen eingeführt.

Zeitbedarf: 18 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Umsetzung von Anforderungen in Entwurfsdiagramme (a) Aus Anforderungsbeschreibungen werden Objekte mit ihren Eigenschaften identifiziert (b) Gleichartige Objekte werden in Klassen (Entwurf) zusammengefasst und um Datentypen und Methoden erweitert	Die Schülerinnen und Schüler • analysieren und erläutern eine objektorientierte Modellierung (A), • stellen die Kommunikation zwischen Objekten grafisch dar (M), • ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M), • modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M), • ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M), • ordnen Klassen, Attributen und Methoden ihren Sichtbarkeitsbereich zu (M), • modellieren Klassen unter Verwendung von Vererbung (M), • implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I), • testen Programme schrittweise anhand von Beispielen (I), • interpretieren Fehlermeldungen und korrigieren den Quellcode (I), • modifizieren einfache Algorithmen und Programme (I), • stellen Klassen, Assoziations- und Vererbungsbeziehungen in Diagrammen grafisch dar (D), • dokumentieren Klassen durch Beschreibung der Funktionalität der Methoden (D).	
2. Implementationsdiagramme als erster Schritt der Programmierung (a) Erweiterung des Entwurfsdiagramms um Konstruktoren und get- und set-Methoden (b) Festelegung von Datentypen in Java, sowie von Rückgaben und Parametern (c) Entwicklung von Klassendokumentationen (d) Erstellung von Sequenzdiagrammen als Vorbereitung für die Programmierung		<i>Beispiele:</i> Buch Seite 96 ff.,
3. Programmierung anhand der Dokumentation und des Implementations- und Sequenzdiagrammes (a) Klassen werden in Java-Quellcode umgesetzt (b) Das Geheimnisprinzip wird umgesetzt (c) Einzelne Klassen und das Gesamtsystem werden anhand der Anforderungen und Dokumentationen auf ihre Korrektheit überprüft.		
4. Vererbungsbeziehungen (a) Das Grundprinzip der Vererbung wird erarbeitet (b) Die Vorteile der Vererbungsbeziehungen (c) Vererbung wird implementiert		<i>Beispiele:</i> Buch Seite 117 ff.,
5. Softwareprojekt (a) Analyse und Dekonstruktion eines Spiels (Modelle, Quelltexte) (b) Erweiterung des Spiels um weitere Funktionalitäten (c) Modellierung eines Spiels aufgrund einer Anforderungsbeschreibung, inklusive einer grafischen Benutzeroberfläche (d) (arbeitsteilige) Implementation des Spiels		<i>Beispiele:</i> Buch Seite 170 ff.,

Unterrichtsvorhaben EF-V (10h)

Thema: Such- und Sortialgorithmen anhand kontextbezogener Beispiele

Leitfragen: *Wie können Objekte bzw. Daten effizient sortiert werden, so dass eine schnelle Suche möglich wird?*

Vorhabenbezogene Konkretisierung:

Dieses Unterrichtsvorhaben beschäftigt sich mit der Erarbeitung von Such- und Sortialgorithmen. Der Schwerpunkt des Vorhabens liegt dabei auf den Algorithmen selbst und nicht auf deren Implementierung in einer Programmiersprache, auf die in diesem Vorhaben vollständig verzichtet werden soll.

Zunächst lernen die Schülerinnen und Schüler das Feld als eine erste Datensammlung kennen. Optional können nun zunächst die wesentlichen Eigenschaften von Algorithmen wie z.B. Korrektheit, Terminiertheit, Effizienz und Verständlichkeit sowie die Schritte einer Algorithmenentwicklung erarbeitet werden (Klärung der Anforderung, Visualisierung, Zerlegung in Teilprobleme).

Daran anschließend lernen die Schülerinnen und Schüler zunächst Strategien des Suchens (lineare Suche, binäre Suche) und dann des Sortierens (Selection Sort, Insertion Sort, Bubble Sort) kennen. Die Projekteinstiege dienen dazu, die jeweiligen Strategien handlungsorientiert zu erkunden und intuitive Effizienzbetrachtungen der Suchalgorithmen vorzunehmen.

Schließlich wird die Effizienz unterschiedlicher Sortierv Verfahren beurteilt.

Zeitbedarf: 9 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Modellierung und Implementation von Datenansammlungen (a) Modellierung von Attributen als Felder (b) Deklaration, Instanziierung und Zugriffe auf ein Feld	Die Schülerinnen und Schüler • beurteilen die Effizienz von Algorithmen am Beispiel von Sortierverfahren hinsichtlich Zeit und Speicherplatzbedarf (A), • entwerfen einen weiteren Algorithmus zum Sortieren (M), • analysieren Such- und Sortieralgorithmen und wenden sie auf Beispiele an (D).	<i>Beispiele:</i> Buch Seite 130 ff.,
2. Explorative Erarbeitung eines Sortierverfahrens (a) Sortierprobleme im Kontext informatischer Systeme und im Alltag (z.B. Dateisortierung, Tabellenkalkulation, Telefonbuch usw.) (b) Vergleich zweier Elemente als Grundlage eines Sortieralgorithmus (c) Erarbeitung eines Sortieralgorithmus durch die Schülerinnen und Schüler		
3. Systematisierung von Algorithmen und Effizienzbetrachtungen (a) Formulierung oder Erläuterung von mehreren Algorithmen im Pseudocode (auf jeden Fall: Einfügen, Auswählen, Bubble) (b) Anwendung von Sortieralgorithmen auf verschiedene Beispiele (c) Bewertung von Algorithmen anhand der Anzahl der nötigen Vergleiche (d) Effizienzbetrachtungen an einem konkreten Beispiel bezüglich der Rechenzeit und des Speicherplatzbedarfs		
4. Binäre Suche auf sortierten Daten (a) Suchaufgaben im Alltag und im Kontext informatischer Systeme (b) Effizienzbetrachtungen zur binären Suche		

Unterrichtsvorhaben EF-VI (15h)

Thema: Leben in der digitalen Welt – Immer mehr Möglichkeiten und immer mehr Gefahren!?

Leitfrage: *Welche Entwicklung durchlief die moderne Datenverarbeitung und welche Auswirkungen ergeben sich insbesondere hinsichtlich neuer Anforderungen an den Datenschutz daraus?*

Vorhabenbezogene Konkretisierung:

Das folgende Unterrichtsvorhaben stellt den Abschluss der Einführungsphase dar. Schülerinnen und Schüler sollen selbstständig informatische Themenbereiche aus dem Kontext der Geschichte der Datenverarbeitung und insbesondere den daraus sich ergebenden Fragen des Datenschutzes bearbeiten. Diese Themenbereiche werden in Kleingruppen bearbeitet und in Form von Präsentationen vorgestellt. Schülerinnen und Schüler sollen dabei mit Unterstützung des Lehrenden selbstständige Recherchen zu ihren Themen anstellen und auch eine sinnvolle Eingrenzung ihres Themas vornehmen.

Darüber hinaus wird beispielhaft analysiert und bewertet, welche Möglichkeiten und Gefahren die moderne Informationsverarbeitung mit sich bringt.

Anschließend wird verstärkt auf den Aspekt des Datenschutzes eingegangen. Dazu wird das Bundesdatenschutzgesetz in Auszügen behandelt und auf schülernahe Beispielsituationen zur Anwendung gebracht. Dabei steht keine formale juristische Bewertung der Beispielsituationen im Vordergrund, die im Rahmen eines Informatikunterrichts auch nicht geleistet werden kann, sondern vielmehr eine persönliche Einschätzung von Fällen im Geiste des Datenschutzgesetzes.

Zeitbedarf: 15 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Selbstständige Erarbeitung von Themen durch die Schülerinnen und Schüler (a) Mögliche Themen zur Erarbeitung in Kleingruppen: - „Eine kleine Geschichte der Digitalisierung: vom Morsen zum modernen Digitalcomputer“ - „Eine kleine Geschichte der Kryptographie: von Caesar zur Enigma“ - „Von Nullen, Einsen und mehr: Stellenwertsysteme und wie man mit ihnen rechnet“ - „Kodieren von Texten und Bildern: ASCII, RGB und mehr“ - „Auswirkungen der Digitalisierung: Veränderungen der Arbeitswelt und Datenschutz“ (b) Vorstellung und Diskussion durch Schülerinnen und Schüler	Die Schülerinnen und Schüler • bewerten anhand von Fallbeispielen die Auswirkungen des Einsatzes von Informatiksystemen (A), • erläutern wesentliche Grundlagen der Geschichte der digitalen Datenverarbeitung (A), • stellen ganze Zahlen und Zeichen in Binärcodes dar (D), • interpretieren Binärcodes als Zahlen und Zeichen (D), • nutzen das Internet zur Recherche, zum Datenaustausch und zur Kommunikation. (K).	<i>Beispiel:</i> Ausstellung zu informatischen Themen Die Schülerinnen und Schüler bereiten eine Ausstellung zu informatischen Themen vor. Dazu werden Stellwände und Plakate vorbereitet, die ggf. auch außerhalb des Informatikunterrichts in der Schule ausgestellt werden können. <i>Materialien:</i> Schülerinnen und Schüler recherchieren selbstständig im Internet, in der Schulbibliothek, in öffentlichen Bibliotheken, usw.
2. Vertiefung des Themas Datenschutz (a) Erarbeitung grundlegender Begriffe des Datenschutzes (b) Problematisierung und Anknüpfung an die Lebenswelt der Schülerinnen und Schüler (c) Diskussion und Bewertung von Fallbeispielen aus dem Themenbereich „Datenschutz“		<i>Beispiel:</i> Fallbeispiele aus dem aktuellen Tagesgeschehen Die Schülerinnen und Schüler bearbeiten Fallbeispiele aus ihrer eigenen Erfahrungswelt oder der aktuellen Medienberichterstattung. <i>Materialien:</i> Materialblatt zum Bundesdatenschutzgesetz (Download EF-VI.1)

2.1.3 Übersichtsraster Unterrichtsvorhaben Qualifikationsphase

Die folgenden Kompetenzen aus dem Bereich Kommunizieren und Kooperieren werden in allen Unterrichtsvorhaben der Qualifikationsphase vertieft und sollen aus Gründen der Lesbarkeit nicht in jedem Unterrichtsvorhaben separat aufgeführt werden:

Die Schülerinnen und Schüler

- verwenden die Fachsprache bei der Kommunikation über informatische Sachverhalte (K),
- nutzen das verfügbare Informatiksystem zur strukturierten Verwaltung von Dateien unter Berücksichtigung der Rechteverwaltung (K),
- organisieren und koordinieren kooperatives und eigenverantwortliches Arbeiten (K),
- strukturieren den Arbeitsprozess, vereinbaren Schnittstellen und führen Ergebnisse zusammen (K),
- beurteilen Arbeitsorganisation, Arbeitsabläufe und Ergebnisse (K),
- präsentieren Arbeitsabläufe und -ergebnisse adressatengerecht (K).

Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase – Q1

Qualifikationsphase – Q1	
<u>Unterrichtsvorhaben Q1-I</u> Thema: Wiederholung und Vertiefung der objektorientierten Modellierung Zentrale Kompetenzen: - Modellieren - Darstellen und Interpretieren - Implementieren - Kommunizieren und Kooperieren Inhaltsfelder: - Daten und ihre Strukturierung - Algorithmen - Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: - Objekte und Klassen - Wirkung der Automatisierung Zeitbedarf: 14 Stunden	<u>Unterrichtsvorhaben Q1-II</u> Thema: Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen Zentrale Kompetenzen: - Modellieren - Implementieren - Darstellen und Interpretieren - Argumentieren Inhaltsfelder: - Daten und ihre Strukturierung - Algorithmen Inhaltliche Schwerpunkte: - Objekte und Klassen - Syntax und Semantik einer Programmiersprache - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten Zeitbedarf: 20 Stunden
<u>Unterrichtsvorhaben Q1-III</u> Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen Zentrale Kompetenzen: - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren Inhaltsfelder: - Algorithmen - Formale Sprachen und Automaten Inhaltliche Schwerpunkte: - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache Zeitbedarf: 20 Stunden	<u>Unterrichtsvorhaben Q1-IV</u> Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten Zentrale Kompetenzen: - Argumentieren - Modellieren - Implementieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren Inhaltsfelder: - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten - Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: - Datenbanken - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache - Sicherheit Zeitbedarf: 20 Stunden
Summe Qualifikationsphase 1: 74 Stunden	

Übersichtsraster Unterrichtsvorhaben in der Qualifikationsphase – Q2

Qualifikationsphase – Q2	
<u>Unterrichtsvorhaben Q2-I</u> Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen Zentrale Kompetenzen: - Argumentieren - Darstellen und Interpretieren - Modellieren - Implementieren - Kommunizieren und Kooperieren Inhaltsfelder: - Daten und ihre Strukturierung - Algorithmen - Formale Sprachen und Automaten Inhaltliche Schwerpunkte: - Objekte und Klassen - Analyse, Entwurf und Implementierung von Algorithmen - Algorithmen in ausgewählten informatischen Kontexten - Syntax und Semantik einer Programmiersprache Zeitbedarf: 20 Stunden	<u>Unterrichtsvorhaben Q2-II</u> Thema: Automaten und formale Sprachen Zentrale Kompetenzen: - Argumentieren - Darstellen und Interpretieren - Modellieren - Kommunizieren und Kooperieren Inhaltsfelder: - Formale Sprachen und Automaten - Informatiksysteme Inhaltliche Schwerpunkte: - Syntax und Semantik einer Programmiersprache - Endliche Automaten - Grammatiken regulärer Sprachen - Möglichkeiten und Grenzen von Automaten und formalen Sprachen Zeitbedarf: 20 Stunden
<u>Unterrichtsvorhaben Q2-III</u> Thema: Aufbau von und Kommunikation in Netzwerken Zentrale Kompetenzen: - Argumentieren - Darstellen und Interpretieren - Kommunizieren und Kooperieren Inhaltsfelder: - Informatiksysteme - Informatik, Mensch und Gesellschaft Inhaltliche Schwerpunkte: - Einzelrechner und Rechnernetzwerke - Sicherheit - Nutzung von Informatiksystemen, Wirkungen der Automatisierung Zeitbedarf: 16 Stunden	
Summe Qualifikationsphase 2: 56 Stunden	

2.1.4 Konkretisierte Unterrichtsvorhaben Qualifikationsphase

Die Umsetzung des Lehrplans folgt pro Unterrichtsvorhaben jeweils auf einer Übersichts- und einer dazugehörigen Konkretisierungsebene. Die Übersichtsebene ist für die Kolleginnen und Kollegen verpflichtend, die hier dargestellten Konkretisierungen beziehen sich auf das Lehrwerk **Informatik 2** und können anhand der Lehrtexte, Projekte und Aufgaben umfassend umgesetzt werden.

Didaktische Lernumgebung

Die Unterrichtsvorhaben in Informatik basieren in ihrer konkreten Umsetzung auf der Programmierumgebung Java.

Zur lehrplankonformen und arbeitssparenden Umsetzung von Modellierung und Programmierung wird der Java Editor (<http://www.javaeditor.org/doku.php>) verwendet.

Unterrichtsvorhaben Q1-I

Thema: Wiederholung und Vertiefung der objektorientierten Modellierung

Leitfragen: Wie wird aus einem anwendungsbezogenen Sachkontext ein informatisches Klassenmodell entwickelt? Wie werden Attribute, Methoden und Beziehungen identifiziert, den Klassen zugeordnet und dargestellt? Welche Auswirkungen hat die informatisch-technische Entwicklung auf das Leben der Menschen?

Vorhabenbezogene Konkretisierung:

Der bereits bekannte objektorientierte Zugang zu informatischer Modellierung wird von einer allgemeinen Betrachtung dieses informatischen Konzepts auf eine konkrete Problematik übertragen. Anhand dieser wird eine anwendungsbezogene Implementation Schritt für Schritt von der Objektidentifikation über das Entwurfs- und Implementationsdiagramm durchlaufen.

Grundlegende Modellierungskonzepte wie Sichtbarkeiten, Assoziationen, Vererbung sowie deren Darstellung in Entwurfs- und Klassendiagrammen und Dokumentationen werden wiederholt. Ebenso wird erneut die grafische Darstellung von Objektkommunikation thematisiert.

Anhand von Gütekriterien und Eigenschaften von Modellierung entwickeln und bewerten die Schülerinnen und Schüler Klassenentwürfe.

Das Konzept der objektorientierten Modellierung wird um die Idee der abstrakten Klasse sowie um das Subtyping erweitert.

Zeitbedarf: 14 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Wiederholung der grundlegenden Konzepte der objektorientierten Programmierung a) Sichtweise der objektorientierten Informatik auf die Welt b) OOP als informatikspezifische Modellierung der Realität c) Schritte der Softwareentwicklung 2. Erweiterung der objektorientierten Programmierung a) Umsetzung einer Anforderung in Entwurfs- und Klassendiagramm b) Objektkommunikation im Sequenzdiagramm c) Klassendokumentation d) Umsetzung von Teilen der Modellierung 3. Übung und Vertiefung der OOM / OOP	Die Schülerinnen und Schüler ... - analysieren und erläutern objektorientierte Modellierungen (A), - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von Multiplizitäten (M), - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M), - modellieren abstrakte und nicht abstrakte Klassen unter Verwendung von Vererbung durch Spezialisieren und Generalisieren (M), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - wenden eine didaktisch orientierte Entwicklungsumgebung zur Demonstration, zum Entwurf, zur Implementierung und zum Test von Informatiksystemen an (I), - stellen Klassen und ihre Beziehungen in Diagrammen grafisch dar (D), - dokumentieren Klassen (D), - stellen die Kommunikation zwischen Objekten grafisch dar (D),	2.1 Die Welt ist voller Objekte Projekteinstieg: Klassenentwurf – step by step 2.2 Gut geplant – Klassenentwurf 2.3 Vererbungshierarchien nutzen Prüfungsvorbereitung

Unterrichtsvorhaben Q1-II

Thema:

Organisation und Verarbeitung von Daten I – Modellierung und Implementierung von Anwendungen mit dynamischen und linearen Datenstrukturen

Leitfragen:

Wie müssen Daten linear strukturiert werden, um in den gestellten Anwendungsszenarien eine beliebige Anzahl von Objekten verwalten zu können?

Vorhabensbezogene Konkretisierung:

Ausgehend von einigen Alltagsbeispielen werden als Erstes die Anforderungen an eine Datenstruktur erschlossen. Anschließend werden die Möglichkeiten des Arrays untersucht, lineare Daten zu verwalten und über deren Grenzen/Probleme die Vorteile einer dynamischen linearen Struktur am Beispiel der Struktur Queue erarbeitet (Anwendungskontext Warteschlange). Die Klasse *Queue* selbst wird vorgegeben, die Operationen erläutert. Zur Vertiefung der Kenntnisse wird ein weiteres Anwendungsszenario eingeführt, dessen Lösung modelliert und implementiert wird. Darauf folgt die Erarbeitung der Struktur Stack, die mithilfe eines einfachen Anwendungsszenarios eingeführt wird. Auch hier wird die Klasse *Stack* selbst vorgegeben und die Operationen erläutert. Weitere Aufgaben dienen der Vertiefung und Sicherung.

Um die Unterschiede der beiden Prinzipien FIFO und LIFO zu verstehen, werden zur Lösung der Aufgaben sowohl der Stack als auch die Queue benötigt.

Als letzte lineare dynamische Datenstruktur wird die Liste eingeführt. In dieser Sequenz liegt der Fokus auf der Möglichkeit, auf jedes Element zugreifen zu können. Nachdem die umfangreicheren Standardoperationen dieser Datenstruktur in einem einführenden Beispiel (Vokabeltrainer) erarbeitet und in einem weiteren Beispiel vertieft wurden, werden abschließend in einem Anwendungskontext verschiedene lineare Datenstrukturen angewendet. Die Modellierung erfolgt beim gesamten Vorhaben in Entwurfs- und Implementationsdiagrammen.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Die Datenstruktur Feld a) Erarbeitung der Anforderungen an eine Datenstruktur b) Wiederholung der Datenstruktur Array, Eigenschaften der Datenstruktur, Standardoperationen für ein und zweidimensionale Arrays c) Modellierung und Implementierung von Anwendungen	Die Schülerinnen und Schüler... - erläutern und begründen methodische Vorgehensweisen, Entwurfs- und Implementationsentscheidungen sowie Aussagen über Informatiksysteme (A) - konstruieren zu kontextbezogenen Problemstellungen informatische Modelle (M) - ermitteln bei der Analyse von Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M) - ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen sowie lineare und nichtlineare Datensammlungen zu (M),	Kapitel: - Anforderungen an eine Datenstruktur - Speichern mit Struktur <i>Beispiele:</i> Buch Seite 52 ff.,
2. Die Datenstruktur Schlange a) Modellierung und Implementierung der Verknüpfung von Objekten b) Generische Typen, Trennung von Verwaltung und Inhalt dynamischer Datenstruktur c) Erläuterung von Problemstellungen, die nach dem FIFO-Prinzip bearbeitet werden d) Funktionalität der Schlange unter Verwendung der Klasse <i>Queue</i>; Erschließen der Standardoperationen e) Modellierung und Implementierung einer Anwendung auf der Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Queue</i>	- implementieren auf der Grundlage von Modellen oder Modellausschnitten Computerprogramme (I) - testen und korrigieren Computerprogramme (I) - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - überführen gegebene textuelle und grafische Darstellungen informatischer Zusammenhänge in die jeweils andere Darstellungsform (D) - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - stellen lineare und nichtlineare Strukturen grafisch dar und erläutern ihren Aufbau (D) - modellieren Klassen mit ihren Attributen, Methoden und ihren Assoziationsbeziehungen unter Angabe von	Kapitel: - Wer zuerst kommt - Objekte miteinander verketten - Verwaltung und Inhalt - Funktionen der Queue Aufgaben: - Warteschlange Büro (Standardoperationen/Basiskompetenz) Kunden warten auf einem Flur, um in ein Büro vorgelassen zu werden. Sie können sich am Ende der Warteschlange anstellen, vorgelassen werden oder müssen alle gehen, wenn die Sprechzeit vorüber ist. <i>Beispiele:</i> Buch Seite 57 ff.,

3. Die Datenstruktur Stapel a) Erläuterung von Problemstellungen, die nach dem LIFO-Prinzip bearbeitet werden b) Funktionalität der Klasse Stapel unter Verwendung der Klasse <i>Stack</i> c) Modellierung und Implementierung einer Anwendung auf Basis einer Anforderungsbeschreibung mit Objekten der Klasse <i>Stack</i> d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (Objekte der Klassen <i>Queue</i>, <i>Stack</i> und <i>Array</i>)	Multiplizitäten (M) - ordnen Klassen, Attributen und Methoden ihre Sichtbarkeitsbereiche zu (M) - dokumentieren Klassen (D) - implementieren Klassen in einer Programmiersprache auch unter Nutzung dokumentierter Klassenbibliotheken (I)	Kapitel: - Daten gut abgelegt – Stapel - Funktionen der Datenstruktur Stapel Aufgaben: - Standardoperationen/ Basiskompetenz (Stapel Münzen/ CDs) zur Umsetzung der gegebenen Funktionen der Klasse <i>Stack</i> - Biber und Teller Es gibt große und kleine Biber sowie grüne und braune Teller. Es muss überprüft werden, ob die gestapelten Teller zur Schlange der Biber passen, da die großen Biber nur von den braunen Tellern essen und die kleinen von den grünen. Hierbei müssen sowohl <i>Queue</i> als auch <i>Stack</i> verwendet werden. - Palindrom Es wird überprüft, ob ein beliebiges Wort ein Palindrom ist.
4. Die Datenstruktur Liste a) Analyse der Möglichkeiten bisheriger Datenstrukturen zwecks Bestimmung notwendiger Funktionalitäten für komplexere Anwendungen (Abgrenzung zu <i>Stack/Queue</i>, zusätzliche Fähigkeiten der Klasse <i>List</i>) b) Erarbeitung der Funktionalität der Liste unter Verwendung der Klasse <i>List</i> c) Modellierung und Implementierung einer Anwendung mit Objekten der Klasse <i>List</i> d) Modellierung und Implementierung einer Anwendung unter Verwendung verschiedener Datenstrukturen (<i>Stack</i>, <i>Queue</i>, <i>List</i>) 5. Übungen und Vertiefungen zur Verwendung linearer und dynamischer Datenstrukturen anhand weiterer Problemstellungen		Kapitel: - Flexibel für alle Fälle – (die) lineare Liste - Funktionen der Datenstruktur Liste Aufgaben: - Vokabeltrainer <i>Beispiele:</i> Buch Seite 70 ff., Prüfungsvorbereitung

Unterrichtsvorhaben Q1-III

Thema: Algorithmen zum Suchen und Sortieren auf linearen Datenstrukturen

Leitfragen: Nach welchen Grundprinzipien können Algorithmen strukturiert werden? Welche Qualitätseigenschaften sollten Algorithmen erfüllen? Wie können mithilfe von Such- und Sortieralgorithmen Daten in linearen Strukturen effizient (wieder-)gefunden werden?

Vorhabenbezogene Konkretisierung:

Zunächst werden anhand eines Anwendungsbeispiels übergreifende Algorithmeigenschaften (wie Korrektheit, Effizienz und Verständlichkeit) erarbeitet und Schritte der Algorithmusentwicklung wiederholt. Dabei kommen Struktogramme zur Darstellung von Algorithmen zum Einsatz.

Als besondere Struktur von Algorithmen wird die Rekursion an Beispielen veranschaulicht und gegenüber der Iteration abgegrenzt. Rekursive Algorithmen werden von den Schülerinnen und Schülern analysiert und selbst entwickelt.

In der zweiten Unterrichtssequenz geht es um die Frage, wie Daten in linearen Strukturen (lineare Liste und Array) (wieder-)gefunden werden können. Die lineare Suche als iteratives und die binäre Suche als rekursives Verfahren werden veranschaulicht und implementiert. Die Bewertung der Algorithmen erfolgt, indem jeweils die Anzahl der Vergleichsoperationen und der Speicherbedarf ermittelt wird.

Möchte man Daten effizient in einer linearen Struktur wiederfinden, so rückt zwangsläufig die Frage nach einer Sortierstrategie in den Fokus. Es wird mindestens ein iteratives und ein rekursives Sortierverfahren erarbeitet und implementiert sowie ihre Effizienz bewertet.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Eigenschaften von Algorithmen a) Qualitätseigenschaften von Algorithmen b) Strukturierung von Algorithmen mit Hilfe der Strategien „Modularisierung“ und „Teile und Herrsche“ c) Analyse und Entwicklung von rekursiven Algorithmen	Die Schülerinnen und Schüler - analysieren und erläutern Algorithmen und Programme (A), - modifizieren Algorithmen und Programme (I), - stellen iterative und rekursive Algorithmen umgangssprachlich und grafisch dar (D), - entwickeln iterative und rekursive Algorithmen unter Nutzung der Strategien „Modularisierung“ und „Teile und Herrschen“ (M),	4.1 Ohne Algorithmen läuft nichts 4.2 Teile die Arbeit – rekursive Algorithmen <i>Beispiele: Türme von Hanoi, Fibonacci-Folge,</i> Buch Seite 95 ff.,
2. Suchen in Listen und Arrays a) Lineare Suche in Listen und Arrays b) Binäre Suche in einem Array c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf	- implementieren iterative und rekursive Algorithmen auch unter Verwendung von dynamischen Datenstrukturen (I), - testen Programme systematisch anhand von Beispielen (I). - implementieren und erläutern iterative und rekursive Such- und Sortierv Verfahren (I), - beurteilen die Effizienz von Algorithmen unter Berücksichtigung des Speicherbedarfs und der Zahl der Operationen (A),	4.3 Suchen – iterativ und rekursiv <i>Beispiele:</i> Buch Seite 108 ff.,
3. Sortieren auf Listen und Arrays a) Entwicklung und Implementierung eines iterativen Sortiervfahrens für eine Liste b) Entwicklung und Implementierung eines rekursiven Sortiervfahrens für ein Array c) Untersuchung der beiden Verfahren bzgl. Laufzeit und Speicherplatzbedarf	- beurteilen die syntaktische Korrektheit und die Funktionalität von Programmen (A), - nutzen die Syntax und Semantik einer Programmiersprache bei der Implementierung und zur Analyse von Programmen (I), - interpretieren Fehlermeldungen und korrigieren den Quellcode (I), - testen Programme systematisch anhand von Beispielen (I),	4.4 Sortieren – iterativ und rekursiv <i>Beispiele:</i> Buch Seite 114 ff.,

Unterrichtsvorhaben Q-I IV

Thema: Nutzung und Modellierung von relationalen Datenbanken in Anwendungskontexten

Leitfragen: Was sind Datenbanken und wie kann man mit ihnen arbeiten? Wie entwickelt man selbst eine Datenbank für einen Anwendungskontext?

Vorhabenbezogene Konkretisierung:

Am Beispiel eines Online-Buchhandels werden der Aufbau einer Datenbank sowie wichtige Grundbegriffe erarbeitet. Die Schülerinnen und Schüler nehmen dabei zunächst die Sicht der Anwender an, die eine bestehende Datenbank beschreiben und analysieren und mithilfe von SQL-Abfragen Daten gezielt herausfiltern.

Mithilfe des Projekteinstiegs „Tabellen“ können bereits zu einem frühen Zeitpunkt des Unterrichtsvorhabens Redundanzen, Inkonsistenzen und Anomalien problematisiert werden.

Nachdem die Lernenden in der ersten Sequenz mit Datenbanken vertraut gemacht wurden, nehmen sie nun die Rolle der Entwickler an, indem sie selbst Datenbanken von Grund auf modellieren und das Modell in ein Relationenschema überführen. Sie arbeiten mit Entity-Relationship-Diagrammen, um Entitäten, zugehörige Attribute, Relationen und Kardinalitäten in Anwendungskontexten darzustellen. Gegebene ER-Diagramme werden analysiert, erläutert und modifiziert.

Der bereits in der ersten Sequenz problematisierte Begriff der Redundanz wird am Ende des Unterrichtsvorhabens wieder aufgegriffen, um die Normalisierung von Datenbanken zu thematisieren. Bestehende Datenbankschemata werden hinsichtlich der 1. bis 3. Normalform untersucht und (soweit nötig) normalisiert.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens:

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Nutzung von relationalen Datenbanken a) Aufbau von Datenbanksystemen und Grundbegriffe - Aufgaben und Eigenschaften eines Datenbanksystems - Erarbeitung der Begriffe Tabelle, Attribut, Attributwert, Datensatz, Datentyp, Primärschlüssel, Datenbankschema - Problematisierung von Redundanzen, Anomalien und Inkonsistenzen b) SQL-Abfragen - Erarbeitung der grundlegenden Sprachelemente von SQL (SELECT(DISTINCT), FROM, WHERE, JOIN) - Analyse und Erarbeitung von SQL-Abfragen (AND, OR, NOT, UNION, AS, GROUP BY, ORDER BY, ASC, DESC, COUNT, MAX, MIN, SUM, Arithmetische Operatoren: +, -, *, /, (...), Vergleichsoperatoren: =, <>, >, <, >=, <=, LIKE, BETWEEN, IN, IS NULL, geschachtelte Select-Ausdrücke) c) Vertiefung an einem weiteren Datenbankbeispiel	Die Schülerinnen und Schüler - erläutern die Eigenschaften und den Aufbau von Datenbanksystemen unter dem Aspekt der sicheren Nutzung (A), - analysieren und erläutern die Syntax und Semantik einer Datenbankabfrage (A), - analysieren und erläutern eine Datenbankmodellierung (A), - erläutern die Eigenschaften normalisierter Datenbankschemata (A), - bestimmen Primär- und Sekundärschlüssel (M), - ermitteln für anwendungsbezogene Problemstellungen Entitäten, zugehörige Attribute, Relationen und Kardinalitäten (M), - modifizieren eine Datenbankmodellierung (M), - modellieren zu einem Entity-Relationship-Diagramm ein relationales Datenbankschema (M), - überführen Datenbankschemata in vorgegebene Normalformen (M), - verwenden die Syntax und Semantik einer Datenbankabfragesprache, um Informationen aus einem Datenbanksystem zu extrahieren (I), - ermitteln Ergebnisse von Datenbankabfragen über mehrere verknüpfte Tabellen (D), - stellen Entitäten mit ihren Attributen und die Beziehungen zwischen Entitäten in einem Entity-Relationship-Diagramm grafisch dar	Kapitel 7 Datenbanken <i>Beispiele: Buch Seite 288 ff.,</i> Wissen speichern und verwalten – Datenbanksysteme Daten anordnen mit Tabellen Beispiele: Adressdaten auf dem Smartphone, Buchhandlung Redundanzen, Anomalien und Inkonsistenzen Projekteinstieg: Tabellen Daten filtern mit SQL Komplexe Filter

2. Modellierung von relationalen Datenbanken a) Datenbankentwurf durch ER-Diagramme - Ermittlung von Entitäten, zugehörigen Attributen, Beziehungen und Kardinalitäten in Anwendungssituationen und Modellierung eines Datenbankentwurfs in Form eines Entity-Relationship-Diagramms - Erläuterung und Erweiterung einer Datenbankmodellierung b) Entwicklung eines relationalen Modells aus einem Datenbankentwurf - Überführung eines Entity-Relationship-Diagramms in ein relationales Datenbankschema inklusive der Bestimmung von Primär- und Fremdschlüsseln c) Normalformen - Überprüfung von Datenbankschemata hinsichtlich der 1. bis 3. Normalform und Normalisierung (um Redundanzen zu vermeiden und Konsistenz zu gewährleisten)	(D), - überprüfen Datenbankschemata auf vorgegebene Normalisierungseigenschaften (D).	7.5 Datenbankentwurf <i>Beispiel: Online-Buchhandel</i> <i>Datenanalyse und Entwurf</i> 7.6 Umsetzung des ER-Modells <i>Entitätsmengen</i> <i>m:n-Beziehungen</i> <i>1:n-Beziehungen</i> <i>1:1-Beziehungen</i> <i>Wiederaufgriff des Projekteinstiegs</i> 7.7 Datenbanken verbessern durch Normalformen
3. Übung und Vertiefung der Nutzung und Modellierung von relationalen Datenbanken		Prüfungsvorbereitung

Unterrichtsvorhaben Q2-I

Thema: Organisation und Verarbeitung von Daten II – Modellierung und Implementierung von Anwendungen mit dynamischen nicht-linearen Datenstrukturen

Leitfragen: Wie können Daten mithilfe von Baumstrukturen verwaltet werden? Wie können mit binären Suchbäumen Inhalte sortiert verwaltet werden und welche Vor- und Nachteile bietet dies?

Vorhabenbezogene Konkretisierung:

Anhand des Anwendungskontextes Spielbäume werden zunächst der generelle Aufbau von Baumstrukturen (auch nicht-binäre) und wichtige Grundbegriffe erarbeitet. Die Darstellung von Bäumen mit Knoten und Kanten wird eingeführt.

Anschließend rückt der Fokus auf die binären Bäume, deren rekursiver Aufbau für die Traversierung der Datenstruktur genutzt wird. Die Preorder-Traversierung wird verwendet, um einen gespeicherten Inhalt in einem Binärbaum zu finden (Tiefensuche).

Ein Anwendungskontext wird mithilfe der Klasse `BinaryTree` (der Materialien für das Zentralabitur in NRW) modelliert und (ggf. in Teilen) implementiert. Dabei wird u. a. die Erzeugung eines Binärbaums mithilfe der beiden Konstruktoren der Klasse `BinaryTree` thematisiert.

Möchte man Daten geordnet speichern, so bietet sich die Struktur des binären Suchbaums an. An Beispielen wird zunächst das Prinzip des binären Suchbaums erarbeitet. Die Operationen des Suchens, Einfügens, Löschens und der sortierten Ausgabe werden thematisiert.

Um Daten in einem Anwendungskontext mithilfe eines binären Suchbaums verwalten zu können, müssen sie in eine Ordnung gebracht werden können, d. h. sie müssen vergleichbar sein. Diese Vorgabe wird mithilfe des Interfaces `Item` realisiert, das alle Klassen, dessen Objekte in einem Suchbaum verwaltet werden sollen, implementieren müssen. Auf diese Weise wird ein Anwendungskontext (Benutzerverwaltung) mithilfe der Klassen `BinarySearchTree` und `Item` modelliert und (ggf. in Teilen) implementiert.

Zeitbedarf: 20 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtsvorhaben Q2-II

Thema: Automaten und formale Sprachen

Leitfragen:

Wie lassen sich reale Automaten durch ein Modell formal beschreiben? Wie kann die Art und Weise, wie ein Computer Zeichen (Eingaben) verarbeitet, durch Automaten dargestellt werden? Welche Eigenschaften besitzen Automaten und was können sie leisten? Wie werden sie dargestellt? Wie werden reguläre Sprachen durch eine Grammatik beschrieben? In welchem Verhältnis stehen endliche Automaten und Grammatiken? Welche Anwendungsfälle können durch endliche Automaten und Grammatiken regulärer Sprachen beschrieben werden und welche nicht?

Vorhabensbezogene Konkretisierung:

Ausgehend von der Beschreibung und Untersuchung realer Automaten wird das formale Modell eines endlichen Automaten entwickelt. Neben dem Mealy-Automaten geht es vor allem um den erkennenden endlichen Automaten. Auf die Erarbeitung der Beschreibung folgt die Modellierung

eigener Automaten und die Untersuchung bestehender, um die Eigenschaften und Grenzen eines endlichen Automaten zu erkennen. Hierbei wird dessen Verhalten auf bestimmte Eingaben analysiert.

An den Themenkomplex *Endliche Automaten* schließt sich die Erarbeitung von Grammatiken regulärer Sprachen an. Die Untersuchung beginnt mit der Erschließung der formalen Beschreibung und wird mit der Entwicklung von Grammatiken zu regulären Sprachen fortgeführt. Hierbei wird auch die Beziehung von Grammatiken regulärer Sprachen zu endlichen Automaten an Beispielen erarbeitet und analysiert. Hierzu gehört auch die Untersuchung, welche Problemstellungen durch endliche Automaten und reguläre Grammatiken beschrieben werden können und welche nicht.

Zeitbedarf: 20 Std.

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

Unterrichtssequenzen	Zu entwickelnde Kompetenzen	Beispiele, Medien, Materialien
1. Endliche Automaten a) Erarbeitung der formalen Beschreibung eines Mealy-Automaten und der Darstellungsformen b) Erarbeitung der formalen Beschreibung eines deterministischen endlichen Automaten (DEA) sowie dessen Darstellungsformen; Erschließung der Fachbegriffe Alphabet, Wort, (akzeptierte) Sprache, Determinismus c) Analyse der Eigenschaften von DEA durch die Modellierung eines Automaten zu einer gegebenen Problemstellung, der Modifikation eines Automaten sowie die Überführung der gegebenen Darstellungsform in eine andere 2. Grammatiken regulärer Sprachen a) Erarbeitung der formalen Beschreibung einer regulären Grammatik (Sprache, Terminal und Nicht-Terminal, Produktionen und Produktionsvorschriften) b) Analyse der Eigenschaften einer regulären Grammatik durch deren Entwicklung und Modellierung zu einer gegebenen Problemstellung.	Die Schülerinnen und Schüler - stellen informatische Modelle und Abläufe in Texten, Tabellen, Diagrammen und Grafiken dar (D) - analysieren und erläutern die Eigenschaften endlicher Automaten einschließlich ihres Verhaltens bei bestimmten Eingaben (A) - ermitteln die Sprache, die ein endlicher Automat akzeptiert (D) - entwickeln und modifizieren zu einer Problemstellung endlicher Automaten (M) - stellen endliche Automaten in Tabellen und Graphen dar und überführen sie in die jeweils andere Darstellungsform (D) - entwickeln zur Grammatik einer regulären Sprache einen zugehörigen endlichen Automaten (M) - analysieren und erläutern Grammatiken regulärer Sprachen (A) - modifizieren Grammatiken regulärer Sprachen (M) - entwickeln zu einer regulären Sprache eine Grammatik, die die Sprache erzeugt (M) - entwickeln zur akzeptierten Sprache eines Automaten eine zugehörige Grammatik (M) - beschreiben an Beispielen den Zusammenhang zwischen Automaten und Grammatiken (D) - zeigen die Grenzen endlicher Automaten und regulärer Grammatiken auf	<i>Beispiele: Buch Seite 132 ff.,</i> Vom realen Automaten zum Modell Projekteinstieg: Getränke- oder Fahrkartenautomat, Schatzsuche oder ein anderer realer Automat Der erkennende endliche Automat Wort und Sprache Grammatiken regulärer Sprachen
3. Übungen und Vertiefungen Verwendung endlicher Automaten und Grammatiken regulärer Sprachen		Prüfungsvorbereitung

Unterrichtsvorhaben Q2-III

Thema: Aufbau von und Kommunikation in Netzwerken

Leitfragen: Was macht menschliche Kommunikation aus? Welchen Stellenwert haben technische/ informatische Hilfsmittel für die Kommunikation? Wie werden Daten in einem Netzwerk zwischen den Kommunikationspartnern übertragen? Wie ist die Arbeitsteilung in Netzwerken gestaltet? Wie kann sicher in Netzwerken kommuniziert werden?

Vorhabenbezogene Konkretisierung:

Ausgehend von alltäglicher Face-to-Face-Kommunikation werden die Grundprinzipien sowie die Bewertungskriterien von Kommunikation erläutert. Das Netzwerk wird als vorteilhafte Kommunikationsstruktur dargestellt und anhand von Topologien und Reichweiten kategorisiert. Ausgehend davon wird der Protokollbegriff entwickelt und anhand des TCP/IP-Schichtenmodells analysiert. Anschließend wird das Client-Server-Prinzip vorgestellt und angewandt.

Sichere Kommunikation in Netzen ist nur dank kryptografischer Verfahren möglich. Stellvertretend werden zwei symmetrische und ein asymmetrisches Verfahren erläutert, angewandt und bewertet.

Zeitbedarf: 16 Stunden

Sequenzierung des Unterrichtsvorhabens: s. nächste Seite

2.2 Grundsätze der fachmethodischen und fachdidaktischen Arbeit

In Absprache mit der Lehrerkonferenz sowie unter Berücksichtigung des Schulprogramms hat die Fachkonferenz Informatik die folgenden fachmethodischen und fachdidaktischen Grundsätze beschlossen. In diesem Zusammenhang beziehen sich die Grundsätze 1 bis 13 auf fächerübergreifende Aspekte, die auch Gegenstand der Qualitätsanalyse sind, die Grundsätze 14 bis 21 sind fachspezifisch angelegt.

Überfachliche Grundsätze:

- 1) Geeignete Problemstellungen zeichnen die Ziele des Unterrichts vor und bestimmen die Struktur der Lernprozesse.
- 2) Inhalt und Anforderungsniveau des Unterrichts entsprechen dem Leistungsvermögen der Schüler/innen.
- 3) Die Unterrichtsgestaltung ist auf die Ziele und Inhalte abgestimmt.
- 4) Medien und Arbeitsmittel sind schülernah gewählt.
- 5) Die Schüler/innen erreichen einen Lernzuwachs. Der Unterricht fördert eine aktive Teilnahme der Schüler/innen.
- 6) Der Unterricht fördert die Zusammenarbeit zwischen den Schülern/innen und bietet ihnen Möglichkeiten zu eigenen Lösungen.
- 7) Der Unterricht berücksichtigt die individuellen Lernwege der einzelnen Schüler/innen.
- 8) Die Schüler/innen erhalten Gelegenheit zu selbstständiger Arbeit und werden dabei unterstützt.
- 9) Der Unterricht fördert strukturierte und funktionale Partner- bzw. Gruppenarbeit.
- 10) Der Unterricht fördert strukturierte und funktionale Arbeit im Plenum.
- 11) Die Lernumgebung ist vorbereitet; der Ordnungsrahmen wird eingehalten.
- 12) Die Lehr- und Lernzeit wird intensiv für Unterrichtszwecke genutzt.
- 13) Es herrscht ein positives pädagogisches Klima im Unterricht.

Fachliche Grundsätze:

- 14) Der Unterricht unterliegt der Wissenschaftsorientierung und ist dementsprechend eng verzahnt mit seiner Bezugswissenschaft.
- 15) Der Unterricht ist problemorientiert und soll von realen Problemen ausgehen und sich auf solche rückbeziehen.
- 16) Der Unterricht folgt dem Prinzip der Exemplarizität und soll ermöglichen, informatische Strukturen und Gesetzmäßigkeiten in den ausgewählten Problemen und Projekten zu erkennen.
- 17) Der Unterricht ist anschaulich sowie gegenwarts- und zukunftsorientiert und gewinnt dadurch für die Schülerinnen und Schüler an Bedeutsamkeit.
- 18) Der Unterricht ist handlungsorientiert, d.h. projekt- und produktorientiert angelegt.
- 19) Im Unterricht werden sowohl für die Schule didaktisch reduzierte als auch reale Informatiksysteme aus der Wissenschafts-, Berufs- und Lebenswelt eingesetzt.
- 20) Der Unterricht beinhaltet reale Begegnung mit Informatiksystemen.

2.3 Grundsätze der Leistungsbewertung und Leistungsrückmeldung

Auf der Grundlage von §13 - §16 der APO-GOST sowie Kapitel 3 des Kernlehrplans Informatik für die gymnasiale Oberstufe hat die Fachkonferenz im Einklang mit dem entsprechenden schulbezogenen Konzept die nachfolgenden Grundsätze zur Leistungsbewertung und Leistungsrückmeldung beschlossen. Die nachfolgenden Absprachen stellen die Minimalanforderungen an das lerngruppenübergreifende gemeinsame Handeln der Fachgruppenmitglieder dar. Bezogen auf die einzelne Lerngruppe kommen ergänzend weitere der in den Folgeabschnitten genannten Instrumente der Leistungsüberprüfung zum Einsatz.

2.3.1 Beurteilungsbereich Klausuren

Verbindliche Absprachen:

Bei der Formulierung von Aufgaben werden die für die Abiturprüfungen geltenden Operatoren des Faches Informatik schrittweise eingeführt, erläutert und dann im Rahmen der Aufgabenstellungen für die Klausuren benutzt.

Instrumente:

- | | |
|-------------------------|---|
| • Einführungsphase | eine Klausur je Halbjahr
Dauer der Klausur: zwei Unterrichtsstunden |
| • Qualifikationsphase 1 | zwei Klausuren je Halbjahr
Dauer der Klausuren: zwei Unterrichtsstunden |
| • Qualifikationsphase 2 | zwei Klausuren im 1. Halbjahr (drei Unterrichtsstunden)
eine Klausur im 2. Halbjahr (drei Zeitstunden) |

Die Aufgabentypen, sowie die Anforderungsbereiche I-III sind entsprechend den Vorgaben in Kapitel 3 des Kernlehrplans zu beachten.

Kriterien

Die Bewertung der schriftlichen Leistungen in Klausuren erfolgt über ein Raster mit Hilfspunkten, die im Erwartungshorizont den einzelnen Kriterien zugeordnet sind.

2.3.2 Beurteilungsbereich Sonstige Mitarbeit

Den Schülerinnen und Schülern werden die Kriterien zum Beurteilungsbereich „sonstige Mitarbeit“ zu Beginn des Schuljahres genannt.

Verbindliche Absprachen der Fachkonferenz

Alle Schülerinnen und Schüler führen in der Einführungsphase in Kleingruppen ein Kurzprojekt durch und fertigen dazu eine Arbeitsmappe mit Arbeitstagebuch an. Dies wird in die Note für die Sonstige Mitarbeit einbezogen.

Leistungsaspekte

Mündliche Leistungen

- Beteiligung am Unterrichtsgespräch
- Zusammenfassungen zur Vor- und Nachbereitung des Unterrichts
- Präsentation von Arbeitsergebnissen
- Referate
- Mitarbeit in Partner-/Gruppenarbeitsphasen

Praktische Leistungen am Computer

- Implementierung, Test und Anwendung von Informatiksystemen unter Verwendung der Lernplattform itslearning

Sonstige schriftliche Leistungen

- Arbeitsmappe und Arbeitstagebuch zu einem durchgeführten Unterrichtsvorhaben
- Lernerfolgsüberprüfung durch kurze schriftliche Übungen
In Kursen, in denen höchstens 50% der Kursmitglieder eine Klausur schreiben, finden schriftliche Übungen mindestens einmal pro Kurshalbjahr statt, in anderen Kursen entscheidet über die Durchführung die Lehrkraft.
Schriftliche Leistungsüberprüfungen dauern circa 20 Minuten und umfassen den Stoff der letzten ca. 4–6 Stunden.
- Bearbeitung von schriftlichen Aufgaben im Unterricht

Kriterien

Die folgenden allgemeinen Kriterien gelten sowohl für die mündlichen als auch für die schriftlichen Formen der sonstigen Mitarbeit.

Die Bewertungskriterien stützen sich auf

- die Qualität der Beiträge,
- die Quantität der Beiträge und
- die Kontinuität der Beiträge.

Besonderes Augenmerk ist dabei auf

- die sachliche Richtigkeit,
- die angemessene Verwendung der Fachsprache,
- die Darstellungskompetenz,
- die Komplexität und den Grad der Abstraktion,
- die Selbstständigkeit im Arbeitsprozess,
- die Präzision und
- die Differenziertheit der Reflexion zu legen.

Bei Gruppenarbeiten auch auf

- das Einbringen in die Arbeit der Gruppe,
- die Durchführung fachlicher Arbeitsanteile und
- die Qualität des entwickelten Produktes.

Bei Projektarbeit darüber hinaus auf

- die Dokumentation des Arbeitsprozesses,
- den Grad der Selbstständigkeit,
- die Reflexion des eigenen Handelns und
- die Aufnahme von Beratung durch die Lehrkraft.

Grundsätze der Leistungsrückmeldung und Beratung

Die Grundsätze der Leistungsbewertung werden zu Beginn eines jeden Halbjahres den Schülerinnen und Schülern transparent gemacht. Leistungsrückmeldungen können erfolgen

- nach einer mündlichen Überprüfung,
- bei Rückgabe von schriftlichen Leistungsüberprüfungen,
- nach Abschluss eines Projektes,
- nach einem Vortrag oder einer Präsentation,
- bei auffälligen Leistungsveränderungen,
- auf Anfrage,
- als Quartalsfeedback und
- zu Eltern- oder Schülersprechtagen.

Die Leistungsrückmeldung kann

- durch ein Gespräch mit der Schülerin oder dem Schüler,
- durch einen Feedbackbogen,
- durch die schriftliche Begründung einer Note oder
- durch eine individuelle Lern-/Förderempfehlung

erfolgen.

Leistungsrückmeldungen erfolgen auch in der Einführungsphase im Rahmen der kollektiven und individuellen Beratung zur Wahl des Faches Informatik als fortgesetztes Grund- oder Leistungskursfach in der Qualifikationsphase.

2.4 Lehr- und Lernmittel

Eingeführte Lehrwerke des Schöningh Verlags

Informatik - Lehrwerk für die gymnasiale Oberstufe - Neubearbeitung

Schülerband 1 Einführungsphase

Schülerband 2 Qualifikationsphase

Programmiersprache

JAVA

Programmierungsumgebungen

Greenfoot

Java Editor

Sonstige Software

Freemind,

MS Access

JFlap

3 Entscheidungen zu fach- und unterrichtsübergreifenden Fragen

Die Fachkonferenz Informatik hat sich im Rahmen des Schulprogramms für folgende zentrale Schwerpunkte entschieden:

Zusammenarbeit mit anderen Fächern

Im Informatikunterricht werden Kompetenzen anhand informatischer Inhalte in verschiedenen Anwendungskontexten erworben, in denen Schülerinnen und Schülern aus anderen Fächern Kenntnisse mitbringen können. Diese können insbesondere bei der Auswahl und Bearbeitung von Softwareprojekten berücksichtigt werden und in einem hinsichtlich der informatischen Problemstellung angemessenem Maß in den Unterricht Eingang finden.

Vorbereitung auf die Erstellung der Facharbeit

Möglichst schon im zweiten Halbjahr der Einführungsphase, spätestens jedoch im ersten Halbjahr des ersten Jahres der Qualifikationsphase werden im Unterricht an geeigneten Stellen Hinweise zur Erstellung von Facharbeiten gegeben. Das betrifft u. a. Themenvorschläge, Hinweise zu den Anforderungen und zur Bewertung. Es wird vereinbart, dass nur Facharbeiten vergeben werden, die mit der eigenständigen Entwicklung eines Softwareproduktes verbunden sind.

Exkursionen

In der Oberstufe wird nach Möglichkeit im Rahmen des Unterrichtsvorhabens „Geschichte der digitalen Datenverarbeitung und die Grundlagen des Datenschutzes“ eine Exkursion zum „Heinz Nixdorf Museums Forum“ in Paderborn durchgeführt. Die außerunterrichtliche Veranstaltung wird im Unterricht vor- und nachbereitet.

4 Qualitätssicherung und Evaluation

Durch Diskussion der Aufgabenstellung von Klausuren in Fachdienstbesprechungen und eine regelmäßige Erörterung der Ergebnisse von Leistungsüberprüfungen wird ein hohes Maß an fachlicher Qualitätssicherung erreicht.

Das schulinterne Curriculum (siehe 2.1) ist zunächst bis 2017 für den ersten Durchgang durch die gymnasiale Oberstufe nach Erlass des Kernlehrplanes verbindlich. Nach Abschluss des Abiturs 2017 werden in einer Sitzung der Fachkonferenz Erfahrungen ausgetauscht und ggf. Änderungen für den nächsten Durchgang der Oberstufe beschlossen, um erkannten ungünstigen Entscheidungen schnellstmöglich entgegenwirken zu können. Die Fachgruppe Informatik wird ggf. eine Beschlussvorlage für die erste Fachkonferenz des folgenden Schuljahres erstellen.